PhaseMP: Robust 3D Pose Estimation via Phase-conditioned Human Motion Prior

Mingyi Shi  
The University of Hong Kong  
myshi@cs.hku.hk

Sebastian Starke  
Meta  
sebastian.starke@mail.de

Yuting Ye  
Meta  
yuting.ye@gmail.com

Taku Komura  
The University of Hong Kong  
taku@cs.hku.hk

Jungdam Won*  
Seoul National University  
jungdam@imo.snu.ac.kr

Figure 1. Our novel human motion prior PhaseMP enables us to further constrain the predictions in both nominal and challenging scenarios, resulting in more natural and stable movements. The figure demonstrates the generated motions in several challenging scenarios, which involve generation from incomplete observation in spatial or temporal domain (left), as well as generation from raw videos where heavy occlusion exists (right).

Abstract

We present a novel motion prior, called PhaseMP, modeling a probability distribution on pose transitions conditioned by a frequency domain feature extracted from a periodic autoencoder. The phase feature further enforces the pose transitions to be unidirectional (i.e. no backward movement in time), from which more stable and natural motions can be generated. Specifically, our motion prior can be useful for accurately estimating 3D human motions in the presence of challenging input data, including long periods of spatial and temporal occlusion, as well as noisy sensor measurements. Through a comprehensive evaluation, we demonstrate the efficacy of our novel motion prior, showcasing its superiority over existing state-of-the-art methods by a significant margin across various applications, including video-to-motion and motion estimation from sparse sensor data, and etc.

1. Introduction

Estimating human poses and motions from real-world observations is a fundamental problem in computer vision with numerous potential applications, including human motion understanding, surveillance, motion capture, and human-computer interaction. Recently, there have been significant improvements in 3D human pose/motion estimation from 2D image/video via deep learning; specifically by deploying a data-driven system trained on a paired dataset [23, 31] to learn a mapping from 2D input images to 3D body motions [42, 47, 25, 57, 66]. However, these algorithms still struggle in challenging scenarios given unconstrained real-world data. For example, highly dynamic movements can cause motion blur, and the blurry appearance of the body and the environment can lead to failures in 2D key-point detection, which is a preprocessing step widely used in many 3D pose estimation algorithms. Moreover, occlusions or a restricted camera field-of-view can result in partial or complete lack of pose information. Al-
though certain approaches have attempted to address this challenge by incorporating temporal consistency or formulating it as a denoising problem in the presence of artificial noise [9, 31, 50], they often struggle when dealing with long-period occlusion lasting beyond one second.

In this paper, we aim to develop a 3D pose/motion estimation algorithm as shown in Fig 1 that can work robustly in challenging scenarios illustrated above. More specifically, we propose a novel motion prior, called PhaseMP, modeling the probability distribution on pose transitions. Our novel motion prior draws inspiration from previous models [49, 38]; however, the key distinction lies in its combination with a frequency domain feature extracted from a periodic autoencoder [56], further enhancing the quality and robustness of 3D pose estimation. Intuitively speaking, the phase features play a role akin to long-term physical momentum, facilitating the generation of smooth and stable movements by enforcing a unidirectional motion (i.e., no backward movement in time). Consequently, phase features contribute to imbuing the resulting motions with greater naturalness, particularly in scenarios where the laws of physics play a significant role. This characteristic extends to challenging scenarios where other input features (e.g., joint location) are only partially observable. The underlying ambiguity arising from insufficient contextual cues can be significantly reduced by enforcing the model to maintain the current momentum.

We demonstrate the effectiveness of our motion prior in solving challenging video-to-motion estimation problems, where a large portion of the body is partially invisible for a long period of time due to occlusions or being out-of-sight. Additionally, we showcase its capabilities in addressing other problems such as denoising 3D motion data or estimating full 3D pose from end-effectors only. Through comprehensive evaluation, we demonstrate that our system outperforms state-of-the-art baselines with a large margin.

The paper’s contributions can be summarized as follows: (1) a novel motion prior, combined with the phase feature, applicable to a wide range of 3D human pose estimation systems, (2) a new optimization framework incorporating phase feature energy, which can work robustly for many challenging scenarios where the observation is incomplete or ambiguous in temporal and spatial domains, (3) a comprehensive evaluation showcasing that our system outperforms existing state-of-the-art methods by a large margin, not only in ordinary scenarios but also in challenging scenarios.

2. Related Work

3D Human Pose Estimation As a tool in understanding the human-centered world, there has been a significant body of work focusing on estimating 3D human pose, which is typically represented by a positional skeleton [42, 20, 1, 47, 52, 10, 22, 24, 75], or parametric mesh model [4, 25, 46, 26, 33, 31, 15], from various real-world observations. These approaches can be categorized into two groups: optimization-based and learning-based methods.

The optimization-based methods typically rely on a known transformation function $f$ (e.g., 3D-to-2D projection), and the 3D poses, as the optimization target $x$, are obtained by optimizing an objective function that encourages the transformation $f(x)$ to be close to the observations, represented in the form of 2D joint positions [13, 2] or body mask images [35]. In SMPLify [4], the optimization can be accelerated by constraining the body state using a linear model SMPL [40], the realism of the optimized poses is further improved by incorporating a learned pose prior [46]. However, accurately modeling real-world transformations [32, 65] can be challenging, and test-time optimization is also computationally expensive. To address these limitations, learning-based methods have emerged as a popular alternative, leveraging data to simplify the pose estimation process. These methods typically require a paired dataset [23, 64, 31] of 3D poses and corresponding observations, and learn an end-to-end mapping from the observations to the 3D poses using different specifications, such as known-camera-projection [42, 47], weak-perspective assumption [25, 48, 58, 59], or kinematic structure [52, 36]. However, these approaches often struggle in accurately predicting some key parameters in the wild environment, and may require significant amounts of supervision [34], which can be constrained by the dataset and network architecture.

In our project, we employ an optimization-based approach to achieve more robust pose estimation in challenging scenarios, such as long-term occlusion, where existing methods still struggle to perform effectively.

3D Human Motion Estimation In addition to estimating pose independently on a frame-by-frame basis, existing methods leverage temporal information to improve estimation accuracy. One popular approach is to encode the pose sequence using neural differential operators. For instance, [47] shows that a dilated temporal convolutional network with a sizeable receptive field significantly outperforms single frame methods. More advanced context-aware sequential encoders, such as graph convolutions [30, 75, 6, 78, 22] and transformers [39, 17, 76, 37, 41], which are commonly used in neutral language processing for encoding the long sequence data with parallel multi-head attention mechanism, have also been developed for this task. Additionally, the fusion of spatial and temporal information [71, 51, 77, 6] can be used to avoid conflicts in the sequential model and improve estimation performance. Moreover, adversarial training [31] between real human motions and predicted motion is another option to enhance the realism of pose sequences.
Another line of work tackles sequential pose estimation with learned powerful generative priors, which incorporate probabilistic models that can capture the nature of human movements; such works are based on mixtures-of-Gaussians [21], pose embeddings [45, 12, 63], neural distance field [61], variational autoencoder [14, 38, 19, 67], and the diffusion model [11, 68]. Given observations, the motion estimation is performed by searching for the most plausible alignment on the motion manifold. HumoR [49] achieves impressive motion generation by modeling the pose transitions instead of modeling the poses directly. Their system can support various types of inputs such as RGB-D videos and 2D/3D joint position sequences. Our system also models the pose transitions but with a particular emphasis on incorporating a frequency domain feature for enhancing the robustness of pose estimation in challenging scenarios.

Measuring physical realism or consistency can also lead to a significant improvement in the accuracy of the predicted pose. For example, a consistent skeleton [60, 52, 8] or physics-inspired metrics [44, 16, 53, 69] such as foot sliding, foot-floor penetration, human-environment interaction [72, 74, 73] can improve temporal coherence and reduce the searching space in generating motions.

Motion Frequency Feature The conversion of motion signals from the time domain to the frequency domain [18] has been utilized for motion editing [5, 27], stylization [62, 70], and compression [3]. Compared to the original data domain, the features in frequency domain often remains consistent over a longer period of time, or between different motion clips, making it useful for synthesizing high quality transition motion [56]. We leverage frequency domain features extracted from a large-scale motion database to accurately estimate human movements in videos, including those frames in which the body is partially or fully occluded.

3. System Overview

Our system predicts the full 3D human motion sequence from incomplete data, such as 2D body landmark sequences extracted from videos, 3D end-effector sequences, or noisy 3D joint position sequences. Our system is composed of the periodic autoencoder to extract the phase feature, the phase-conditioned motion prior, and the run-time optimization module. Given a large motion database, the periodic autoencoder is first trained (see Sec 4.1). Using the extracted phase features as additional inputs, a motion prior based is trained (see Sec 4.2, 4.3). At inference time, given an input observation sequence, the 3D human motion is computed via optimization with energy functions that ensure both accuracy and realism (see Sec 5).

4. Phase-conditioned Motion Prior

We first describe how we compute the phase features using a periodic autoencoder trained with a large motion database. Then, we explain the structure of phase-conditioned motion prior and the loss functions to train it.

4.1. Deep Phase Feature

The periodic autoencoder (PAE) [56], which we use to compute the phase features, is depicted on the left side of Figure 2. It is an autoencoder equipped with convolution
and FFT (fast Fourier transform) layers in its intermediate structure, allowing it to compute embeddings in the frequency domain given joint velocities as inputs. Intuitively speaking, it learns alignments of periodic signals existing in motions, the learned embedding can play a role of physical momentum as a result. More specifically, at frame $t$, PAE uses a window of 3D joint velocities $X_t \in \mathbb{R}^{J \times T \times N}$ as input, where $J, N$ represent the number of joints and the size of window, respectively. Given $X_t$ for the frame $t$, the encoding process includes a sequence of 1D convolutions followed by a differentiable FFT layer:

$$A_t, B_t, F_t = \text{FFT}(\text{Conv}(X_t)),$$

where $A_t$, $B_t$, and $F_t$ are amplitudes, offsets, and frequencies, and phase shift $S_t$ of periodic embeddings are obtained by a separate fully-connected network:

$$(s_x, s_y) = \text{FC}(\text{Conv}(X_t)), S_t = \text{atan2}(s_y, s_x),$$

where $\text{atan2}$ is a 2-argument arc-tangent. Then $A_t, B_t, F_t, S_t$ are used to compose phase features in the decoding process, which is conducted by first reconstructing the feature maps $F_t$ in the temporal domain:

$$F_t = A_t \cdot \sin(2\pi \cdot (F_t \cdot T - S_t)) + B_t$$

where $T$ is a known time window. This is followed by a 1D deconvolution $X'_t = \text{DeConv}(F_t)$ for reconstructing the original signals. The entire PAE is trained using the reconstruction loss:

$$L_{PAE} = MSE(X_t, X'_t)$$

Once PAE is trained, the deep phase feature $P_t$ are computed as follows using the frequency domain parameters:

$$P_t = [p_t, F_t, A_t].$$

where $p_t = (A_t \cdot \sin(2\pi \cdot S_t), A_t \cdot \cos(2\pi \cdot S_t))$ is called the phase manifold vectors that periodically change over time, and the last two variables are frequency and amplitude.

### 4.2. Motion Prior Modeling

The right side of Figure 2 illustrates our novel phase-conditioned motion prior. The structure is inspired by MotionVAE [38] and HuMoR [49]; both models, including ours, are based on conditional VAEs [29] where the distribution of plausible pose transition is learned. The key differences are that our model is conditioned by the deep phase feature $P_t$ extracted from pre-trained periodic autoencoder [56] in addition to the pose feature, and sinusoidal activation layers [54] are also incorporated to fully utilize the periodic nature of our phase feature.

For the pose feature, we use the same representation used in HuMoR [49], which includes position, orientation, and corresponding velocities for all the joints. Our motion prior receives the previous phase feature $P_{t-1}$, the previous pose feature $x_{t-1}$, and the next pose feature $x_t$ as input, then predicts the change of pose $\Delta x$ and phase $\Delta P$ features as the output, from which the features are updated as follows:

$$\hat{x}_t = x_{t-1} \bigoplus \Delta x$$

$$\hat{P}_t = P_{t-1} + \Delta P,$$

where $\bigoplus$ is a differentiable integration operator to compute the current pose given the previous pose and its change $\Delta x$, where positional and rotational components are updated via addition and multiplication, respectively.

In this process, $\Delta P$ and $\Delta x$ are computed from a decoder $D$ conditioned on a latent variable (i.e., embedding) $z$ that describes the possible pose transition. We use a learnable prior $R$ and an encoder $E$ (i.e., posterior) similarly to HuMoR [49]. They are defined as follows:

$$z'_{t-1} = \epsilon(R(x_{t-1}, P_{t-1}))$$

$$z_{t-1} = \epsilon(E(x_{t-1}, x_t, P_{t-1}))$$

$$\Delta x, \Delta P, c_t = D(x_{t-1}, z_{t-1}, P_{t-1}),$$

where $\epsilon$ denotes a re-parameterization operation [28], $z'_{t-1}$ and $z_{t-1}$ represents the latent variables sampled from the Gaussian distribution, generated from the prior and the encoder, respectively, and $c_t$ is the contact label to further enhance motion quality.

### 4.3. Prior Training

The pose in the next frame is predicted by sampling a Gaussian distribution produced by the prior model. In the training stage, following the CVAE [55], as shown in the right side of Figure 2, the prior model $R$ is trained with posterior $E$ by aligning their distribution $N(\mu_\theta, \sigma_\theta^2)$ and $N'(\mu_\phi, \sigma_\phi^2)$.

Given the $D$ set of training data in the form of $\{x_{t-1}, P_{t-1}, x_t, P_t\}_t^{D}$, our motion prior is trained by the following loss function composed of four terms:

$$L = L_{rec} + \lambda_{KL}L_{KL} + \lambda_{ct}L_{ct} + \lambda_{SMPL}L_{SMPL}.$$  

with weights $\lambda_{KL} = 4e^{-4}, \lambda_{ct} = 0.01, \lambda_{SMPL} = 0.5$. The first term $L_{rec}$ is the reconstruction loss that minimizes the difference between the decoder output and the ground truth:

$$L_{rec} = ||\hat{x}_t - x_t||^2 + 0.1 \times ||\hat{P}_t - P_t||^2,$$

where $\hat{x}, \hat{P}$ are the predicted pose and phase features, respectively. The second term $L_{KL}$ enforces the distribution learned by the learnable prior is close to the one from the encoder by measuring their KL-divergence. The third term $L_{ct}$ is the contact loss

$$L_{ct} = \sum_j BCE(c'_t, c_t) + c'_t ||c'_t||^2$$
where $BCE$ is the binary cross-entropy; $\hat{c}_t^j$ and $\hat{v}_t^j$ represent the predicted contact label and the velocity of the $j$-th joint, respectively. This term encourages the decoder to predict correct contact labels while keeping the velocities of the joints in contact with the ground to become zero. The final term $L_{SMPL}$ is optional which enforces the generated body meshes to be consistent with the SMPL model in the ground truth dataset, which is the same as one described in HuMoR [49].

4.4. Robust Inference of PhaseMP

Given an initial pose and phase features $x_0$ and $P_0$, an embedding for transition $z_0$ is sampled randomly from the prior distribution $R(x_0, P_0)$, from which the decoder generates the change of pose and phase features, and then they are updated via Eq. 6. This process can be performed repeatedly to generate a continuous motion $\langle x_0, x_1, x_2, \cdots \rangle$. Although we observe that our phase-conditioned motion prior can generate better quality motions already when compared to other methods, the generated motions can still deteriorate especially when the input observation is highly unreliable due to noisy or missing joints. We thus propose a novel way to update the phase feature that considers the confidence of the input observation dynamically. It is computed as follows:

$$
\bar{p}_t = A_t \cdot I(\alpha_F) \cdot (R(\theta) \cdot p_{t-1}, (p_{t-1} + \Delta p))
$$

$$
A_t = (1 - \alpha_A)A_{t-1} + \alpha_A(A_{t-1} + \Delta A)
$$

$$
\bar{F}_t = (1 - \alpha_F)F_{t-1} + \alpha_F(F_{t-1} + \Delta F)
$$

$$
\hat{\theta} = \Delta t \cdot 2\pi \cdot \hat{F}_t
$$

where the update is basically performed by the interpolation of two sources, one from network prediction and the other from a cyclic update by rotating the phase manifold vector $p_{t-1}$ with $\theta$. $I(\cdot)$ refer to linear interpolation, $R(\cdot)$ is the rotation operation, $\Delta t$ is a time-step between adjacent frames, and $\alpha_F \in [0, 1]$ represents how reliable each component of the phase feature is given the current pose feature. The confidence values $\alpha_A$, $\alpha_P$, and $\alpha_F$ are computed by:

$$
\alpha_A, \alpha_P, \alpha_F = \begin{cases} 
0, 0, 0 & \text{if } \phi_t < \phi_{low} \\
\phi_t, \phi_t, \phi_t & \text{if } \phi_{low} \leq \phi_t < \phi_{high} \\
1, 0, 5, 1 & \text{if } \phi_t \geq \phi_{high}
\end{cases}
$$

where $\phi_t \in [0, 1]$ is the confidence value of the input observation at frame $t$ (e.g., values given by 2D pose detectors), and $\phi_{low}$, $\phi_{high}$ are user-specified minimum and maximum confidence thresholds where we use 0.4, 0.8, respectively. Note we set $\alpha_P = 0.5$ even when the input observation is fully reliable as demonstrated in [56]. If the confidence value is high ($\alpha_A = 1$) then the phase feature is updated following Eq. 11. If the confidence is low ($\alpha_A = 0$), the frequency and amplitude feature $A_t, F_t$ are carried over from the previous frame as the same as $A_{t-1}, F_{t-1}$, and are used to update the phase manifold features. Otherwise, the values are interpolated based on corresponding confidence values.

Implementation

The prior and encoder networks are both designed as fully-connected 5-layer MLPs equipped with ReLU activation units and group normalization. The decoder is designed as a 4-layer SirenMLP [54] with sine activation of 60 sine factor. We follow the initialization scheme introduced in Siren [54]. Moreover, we introduce skip connections from the phase feature to each layer of the decoder network, further enhancing its influence. Following MotionVAE [38], we also use scheduled sampling to ensure the network learns from its own predictions. More details will be introduced in the supplemental materials.

5. Test-time Optimization

During run-time, given partial observations, such as 2D landmark sequence or partial 3D joint sequence, we estimate the original 3D pose sequence by optimization. In this section, we outline the details for integrating our phase-conditioned motion prior with optimization.

There are three stages to perform the optimization. Given a sequence of observations $o_{1:T}$ (e.g., a sequence of 2D joint positions in a video) with length $T$, a ground plane is estimated and a sequence of SMPL poses are roughly fitted to the observations in Stage 1. An initial sequence of phase features $P_{1:T}$ is then computed from these poses in Stage 2 using the encoder in Eq. 7. In Stage 3, starting from the roughly fitted poses in Stage 1 and the initialized sequence of phase features in Stage 2, a refinement step is performed by using our PhaseMP to optimize the pose sequence with energies measuring plausibility of the pose sequence. All the above stages are performed by different optimization targets and energy terms.

We will describe Stage 3 here and explain the other stages in the supplementary materials. The optimization
problem in Stage 3 is formulated as follows:

\[
\arg\min_{z_{1:T-1}, \beta, g} (\lambda_{\text{obs}} E_{\text{obs}} + \lambda_{\text{prior}} E_{\text{prior}} + \lambda_{\text{reg}} E_{\text{reg}} + \lambda_{\text{phase}} E_{\text{phase}}) 
\]

where \(\beta\) is the SMPL shape parameter, and \(g\) is the ground plane parameter. The energy function comprises four terms where the first three terms are those proposed in Hu-MoR [49] while the last term is newly introduced in this work, which can significantly improve the motion quality especially for challenging scenarios. The weights can be set differently depending on the types of tasks, our settings for each task are included in Appendix. The details of each energy are explained below.

**Observation Energy** The purpose of the observation energy is to enforce the predicted sequence to align with the given observations:

\[
E_{\text{obs}} = \sum_{t=1}^{T} ||O(\hat{x}_t) - o_t||^2 
\]

where \(o_t\) is an instance of observation at frame \(t\) and \(O\) is a function that projects the pose sequences \(\hat{x}_{0:T}\) which are predicted by our model to the observation space. For instance, a 3D-to-2D projection can be used for the video-to-motion task while a masking function can be used for reconstructing full 3D poses from 3D partial markers.

**Motion Prior Energy** The goal of the motion prior term is to measure whether the given sequence of latent variables \(z_{1:T}\) represents a plausible motion. This can be computed by

\[
E_{\text{prior}} = -\sum_{t=1}^{T-1} \log N(z_t; \mu_0^t, \sigma_0^t) 
\]

where \(\mu_0^t, \sigma_0^t = \mathcal{R}(x_t, P_t)\)

**Regularization Energy** The regularization energy helps the optimized motion to be smooth and consistent. It consists of four terms:

\[
E_{\text{reg}} = \sum_{t=1}^{T} (||\hat{l}_t - \hat{l}_t^{\text{smpl}}||^2 + ||\hat{l}_t - \hat{l}_{t-1}||^2 + \epsilon^2 \hat{l}_t ||\hat{l}_t^{\text{foot}}||^2 + ||g - g_{\text{init}}||^2) 
\]

where the first term regularizes the distance between the predicted joint positions \(\hat{l}_t\) and the joint positions \(\hat{l}_t^{\text{smpl}}\) computed from the SMPL pose parameters; the second term enforces the bone lengths \(l_t\) to be consistent over time; the third term makes the foot stationary by minimizing its velocity \(v^{\text{foot}}\) when the contact label \(c^{\text{foot}}\) is enabled; and the final term prevents the ground plane to deviate from its initial guess \(g_{\text{init}}\) during the optimization.

**Phase-based Energy** As demonstrated in previous work [49], natural-looking motions can be often generated when the input observation is dense and reliable, for example, when all joints are clearly visible in the video and the motion is not extremely dynamic. However, the quality of generated motions is significantly degraded when insufficient cues are provided, due to occlusions or the body being out-of-sight. Simply increasing the weights of the motion prior energy cannot handle invisibility for a long duration (>1s). Here, we introduce a novel phase-based energy to mitigate this challenge.

To calculate the energy, we first compute target phase features \(\bar{P}_{1:T}\) by setting the initial phase to 0 and updating it by inference with Eq. 11. We then optimize the phase features \(P_{1:T}\) by minimizing its difference with the target phase features \(\bar{P}_{1:T}\):

\[
E_{\text{phase}} = \sum_{t=1}^{T} ||\bar{P}_t - P_t||^2. 
\]

For the optimization, \(P_{1:T}\) is first initialized by the phase features extracted from the SMPL poses in Stage 2. This energy can be considered as an additional regularization term in the frequency domain by encouraging the generated motion to maintain similar periodicity to ones computed from reliably observations only.

**Implementation** By default, we use L-BFGS optimization with a step size of 1 and a maximal number of iterations per optimization step of 20, which is implemented by PyTorch. It takes approximately 6 minutes to fit a 3-second sequence with a GTX 3090 graphics card. More details about optimization exist in the supplemental materials.

6. Evaluation

We evaluate our system on (i) motion reconstruction task, (ii) motion completion from sparse 3D joint markers, (iii) video-to-motion task in challenging scenes, (iv) ablation study. More qualitative and quantitative experiments are available in the supplemental materials.

6.1. Datasets and Metrics

**Datasets** The evaluation of our proposed method utilizes three human motion datasets. (1) AMASS [31], the largest dataset in terms of motion capture data, is curated from various sources and represented in the SMPL format, all the
Table 1. Comparison of different methods on the per-frame prediction (Left) and random sampling with different durations (Middle, Right). The MPJPE and DE are measured as positional errors with the unit of centimeters. For the sampling experiments, we use the same initial pose from ground truth, and then run the sampling 50 times, and choose the one with the lowest ADE (average) to report its FDE (final frame).

<table>
<thead>
<tr>
<th>Model</th>
<th>Contact ↑</th>
<th>MPJPE↓</th>
<th>PJPE-std↓</th>
<th>MV-PE↓</th>
<th>Contact ↑</th>
<th>ADE ↓</th>
<th>FDE ↓</th>
<th>Contact ↑</th>
<th>ADE ↓</th>
<th>FDE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>HuMoR [49](MLP, w/o Phase)</td>
<td>0.9770</td>
<td>0.022</td>
<td>0.051</td>
<td>0.057</td>
<td>0.8585</td>
<td>36.14</td>
<td>47.34</td>
<td>0.8216</td>
<td>45.43</td>
<td>62.47</td>
</tr>
<tr>
<td>Ours(MLP, with Phase)</td>
<td>0.9764</td>
<td>0.020</td>
<td>0.040</td>
<td>0.061</td>
<td>0.8646</td>
<td>32.36</td>
<td>35.73</td>
<td>0.8525</td>
<td>39.48</td>
<td>54.96</td>
</tr>
<tr>
<td>Ours(SirenMLP, w/o Phase)</td>
<td>0.9788</td>
<td>0.019</td>
<td>0.031</td>
<td>0.044</td>
<td>0.8691</td>
<td>31.81</td>
<td>35.59</td>
<td>0.8577</td>
<td>35.47</td>
<td>49.95</td>
</tr>
<tr>
<td>Ours(SirenMLP, with Phase)</td>
<td><strong>0.9799</strong></td>
<td><strong>0.017</strong></td>
<td><strong>0.021</strong></td>
<td>0.047</td>
<td><strong>0.8702</strong></td>
<td>34.88</td>
<td>36.91</td>
<td><strong>0.8662</strong></td>
<td>42.12</td>
<td><strong>49.47</strong></td>
</tr>
</tbody>
</table>

Baseline and Metrics  Based on our inspiration from the HuMoR baseline, we propose several enhancements in our system, including a conditional feature, a new network module, and different optimization energy terms. We evaluate the effectiveness of these improvements in challenging scenarios through experiments.

To assess the accuracy of our reconstruction, we compute the commonly used metrics MPJ-PE and MV-PE [42, 31, 49], to show the mean positional distance (cm) of body joints and vertices between our prediction and the ground truth. Additionally, we use the displacement distance (DE) between the generated motion and ground truth, to evaluate whether our learned prior can effectively reconstruct the desired motion. To examine the physical realism of the motion estimation task [38, 49], we also measure the accuracy of contact (Contact), mean per-joint accelerations (Accel), foot-ground penetrations with a 15cm threshold, penetration occurrence frequency (P-Frep), mean penetration distance (P-Dis).

6.2. Evaluation of Motion Reconstruction

Because our model is based on conditional VAEs, we first evaluate our model by measuring the accuracy on reconstruction (prediction). The accuracy is evaluated by two criteria, where we use the AMASS dataset to train and test our model. The first criterion is the per-frame prediction accuracy, where the ground truth input of \( x_{t-1}, P_{t-1} \), and the latent variable \( z_t \) obtained from \( E(x_{t-1}, x_t) \) are given. The second criterion evaluates the sequential output by initializing the system with \( x_0 \) and \( P_0 \) from a test motion, and then sampling 50 different motion sequences autoregressively from the same seed for the same length as the test motion. We select the sequence with the lowest average displacement error (ADE) as the closest prediction. We report both the ADE and last frame distance (FDE) for the selected sequence. The results are presented in Table 1 which shows that our full method (SirenMLP+Phase) achieves the best performance on average and all other variants also outperform the SOTA baseline.

6.3. Estimation from 3D Observations

We conduct an experiment to evaluate the effectiveness of test-time optimization in filling incomplete 3D joints. To simulate real-world occlusions, we generate three types of inputs: 1) occluded joints in time (missing frames) or in space (joints above 0.9m in height are only visible; or the end-effectors are only visible); 2) joint positions with noise. We then recover the original pose sequence using our PhaseMP. The performance is evaluated based on the average positional error for visible (Vis) and occluded (Occ) joints.

The experimental results are presented in Table 2 and Table 3. Our approach outperforms other methods in predicting occluded joints, as demonstrated by the average positional error of all and invisible (Occ) joints. Furthermore, our method can predict contact labels more accurately than HuMoR. Regarding the denoising experiments, our approach produces smoother results, particularly when the degree of noise is high. However, an interesting observation is that our method does not align visible joints as accurately as VPose-r-t. This can be attributed to the additional constraints imposed by the periodic manifold, which encourages more realistic motion rather than focusing solely on local alignments.

Some visualization comparison is shown in Figure 5 where our method generates more natural-looking and dynamic motions with less foot sliding when compared to HuMoR. The differences are best seen in the supplemental video.
Table 2. Comparison of different methods on different input settings: 1) Incomplete joints; 2) Root and 5 end-effectors; 3) Keyframes with 10 frame intervals. Comparisons are performed with two different durations (3 and 5 seconds). All the results are measured by two noise levels are tested.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input Conditions</th>
<th>Vis</th>
<th>Occ</th>
<th>All</th>
<th>Contact↑</th>
<th>Accel</th>
<th>P-Frep</th>
<th>P-Dis</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPoser-t</td>
<td></td>
<td>0.67</td>
<td>20.76</td>
<td>9.22</td>
<td>-</td>
<td>5.71</td>
<td>16.77%</td>
<td>2.28</td>
</tr>
<tr>
<td>MVAE</td>
<td></td>
<td>2.39</td>
<td>19.15</td>
<td>9.52</td>
<td>-</td>
<td>7.12</td>
<td>3.15%</td>
<td>0.30</td>
</tr>
<tr>
<td>HuMoR</td>
<td></td>
<td>1.46</td>
<td>17.40</td>
<td>8.24</td>
<td>0.89</td>
<td>5.38</td>
<td>3.31%</td>
<td>0.26</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>3.94</td>
<td>15.63</td>
<td>8.31</td>
<td>0.89</td>
<td>4.58</td>
<td>3.04%</td>
<td>0.28</td>
</tr>
<tr>
<td>HuMoR</td>
<td></td>
<td>3.05</td>
<td>4.12</td>
<td>3.83</td>
<td>0.96</td>
<td>4.91</td>
<td>0.31%</td>
<td>1.03</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>3.16</td>
<td>4.07</td>
<td>3.79</td>
<td>0.97</td>
<td>4.88</td>
<td>0.28%</td>
<td>1.02</td>
</tr>
<tr>
<td>HuMoR</td>
<td></td>
<td>0.96</td>
<td>7.72</td>
<td>7.49</td>
<td>0.91</td>
<td>1.57%</td>
<td>1.90</td>
<td>11.04</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>3.19</td>
<td>4.92</td>
<td>4.33</td>
<td>0.93</td>
<td>6.33</td>
<td>1.31%</td>
<td>1.72</td>
</tr>
<tr>
<td>1D-Filter</td>
<td>4cm</td>
<td>3.91</td>
<td>-</td>
<td>-</td>
<td>2.45%</td>
<td>0.14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VPoser-t</td>
<td>4cm</td>
<td>3.67</td>
<td>-</td>
<td>-</td>
<td>1.35%</td>
<td>0.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MVAE</td>
<td>4cm</td>
<td>2.68</td>
<td>-</td>
<td>-</td>
<td>1.75%</td>
<td>0.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HuMoR</td>
<td>4cm</td>
<td>2.27</td>
<td>0.97</td>
<td>-</td>
<td>1.18%</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours(w/o Phase)</td>
<td>4cm</td>
<td>2.12</td>
<td>0.97</td>
<td>-</td>
<td>1.22%</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours(with Phase)</td>
<td>4cm</td>
<td>1.96</td>
<td>0.98</td>
<td>-</td>
<td>1.14%</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1D-Filter</td>
<td>12cm</td>
<td>11.89</td>
<td>-</td>
<td>-</td>
<td>4.87%</td>
<td>2.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HuMoR</td>
<td>12cm</td>
<td>34.08</td>
<td>0.77</td>
<td>7.29%</td>
<td>5.26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>12cm</td>
<td>9.42</td>
<td>0.89</td>
<td>4.20%</td>
<td>0.48</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Motion estimation from noisy inputs, where two different noise levels are tested.

6.4. Estimation from RGB Observations

We evaluate the proposed method in the video-to-motion task, where the goal is to estimate 3D pose sequences from RGB videos. We use i3DB [43] and PROX [16] datasets, and also obtain the 2D poses and confidence values from OpenPose [7]. To mitigate the local minima issue, we follow the same procedure proposed by HuMoR, which splits the entire motion into 3-second sub-sequences. However, instead of stitching all sub-sequences after the parallel optimization, we optimize them one by one, using the last frame in the optimized sequence \( s_{t-1} \) as the initial pose of \( s_t \). As a result, we can easily stitch them sequentially to obtain a full-frame reconstruction. The results are presented in Table 4. It is observed that optimization-based methods produce more accurate results, and our proposed method further improves the estimation compared to others. We also show qualitative results in Figure 4. Our method maintains consistency in long sequences thanks to the temporal features captured by the phase feature, which stabilizes the estimation in heavily occluded frames, such as sitting on a sofa. More results for the video-to-motion task can be found in our supplementary materials.

6.5. Ablation Study

Here we analyze the effectiveness of each newly-added component of our system for the motion reconstruction task. The optimization without any new component is used as the baseline. Then we compare the results of the reconstructed motion w/wo SirenMLP (SI), w/wo Phase Condition (PC), w/wo Dynamic Test-Time Interpolation(DI). The experiment is done with i3DB dataset in the same setting described in Sec. 6.4. The results are shown in Tab. 5.

Table 4. Comparison of different methods for the video-to-motion task on i3DB [43] and PROX [16] datasets. The P-MPJPE is used for calculating the positional error after root alignment.

<table>
<thead>
<tr>
<th>Method</th>
<th>i3DB</th>
<th>PROX</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIBE</td>
<td>116.46</td>
<td>7.98% 3.01</td>
</tr>
<tr>
<td>VPoser-t</td>
<td>22.73</td>
<td>16.62 2.68</td>
</tr>
<tr>
<td>MVAE</td>
<td>40.91</td>
<td>19.17 7.43% 1.55</td>
</tr>
<tr>
<td>HuMoR</td>
<td>28.15</td>
<td>14.51 2.12% 0.68</td>
</tr>
<tr>
<td>Ours</td>
<td>27.43</td>
<td>14.19 2.03% 0.61</td>
</tr>
</tbody>
</table>

Table 5. Comparison of different system components involved setting.

The ablation study reveals the effectiveness of different components in our framework. Phase is a powerful signal which helps to reproduce realistic movements. However, using it solely will result in an unstable prediction under heavy occlusion scenarios due to the error accumulation through the auto-regressive phase update. Dynamic interpolation masks out the low confidence frames, and its combination with the phase makes the prediction more robust in challenging cases. The SirenMLP module also improves the baseline results.
Figure 4. Qualitative comparison for video-to-motion task. We run video-based motion reconstruction method VIBE [31], test-time optimization with HuMoR [49] and our method in the same settings and report the results. All the methods are trained with AMASS [31] dataset only. Our method can produce more coherent and realistic motion even in the existence of heavy occlusion.

Figure 5. Motion generation from incomplete input. The first and second rows are the fully body reconstruction from the upper body only with/without using the phase feature. The bottom two rows show motion-in-betweening results from sparse keyframes shown as purple color.

7. Discussion

We have demonstrated the benefits of our phase-conditioned motion prior model, PhaseMP, in a variety of 3D pose estimation tasks under challenging settings. It is a general motion representation model that can be applied to encode not only periodic actions such as locomotion but also complex non-periodic actions such as dancing.

Our system has several limitations. Firstly, it relies on an assumption the motions are performed on flat ground and the cameras are static. Secondly, our system might fail when the occlusion period is excessively long because the ambiguity in the phase prediction increases. Thirdly, though most movements can be reconstructed with the periodic autoencoder, the diversity of the output motions may be restricted to those observed in the training data. Lastly, as with other methods, our optimization process can still encounter failures when tested on motions that significantly deviate from the training data.

One intriguing future direction could involve utilizing the learned phase feature as a positional encoding for a transformer architecture, allowing the phase feature to directly influence predictions without test-time optimization. Another promising avenue for future research would be exploring the integration of multi-modal signals that can be easily combined in the frequency domain, such as sound waves from input videos.

Acknowledgments

This work was mainly done during Mingyi Shi’s internship in Meta. We thank Deepak Gopinath for his valuable input. Jungdam Won was partially supported by the New Faculty Startup Fund from Seoul National University, ICT(Institute of Computer Technology) at Seoul National University, and the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2020-0-01460) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). Taku Komura and Mingyi Shi are partly supported by Technology Commission (Ref:ITS/319/21FP) and Research Grant Council (Ref: 17210222), Hong Kong.
References


Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3d people in depth. In *CVPR*, 2022.


